

Docket No.: 67471-024

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of : Customer Number: 20277
:
Hajime OGAWA, et al. : Confirmation Number:
:
Serial No.: : Group Art Unit:
:
Filed: : Examiner:
:
For: INSTRUCTION SCHEDULING METHOD, INSTRUCTION SCHEDULING DEVICE, AND
INSTRUCTION SCHEDULING PROGRAM

**CLAIM OF PRIORITY AND
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT**

Mail Stop CPD
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 35 U.S.C. 119, Applicants hereby claims the priority of:

Japanese Patent Application No. JP2002-241877, filed on August 22, 2002.

cited in the Declaration of the present application. A certified copy is submitted herewith.

Respectfully submitted,

MCDERMOTT, WILL & EMERY



Ramyar M. Farid
Registration No. 46,692

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 RMF:gav
Facsimile: (202) 756-8087
Date: August 22, 2003

67471-024

Hajime OGAWA, et al.

日 本 国 特 許 庁 August 22, 2003

JAPAN PATENT OFFICE

McDermott, Will & Emery

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 8月22日

出 願 番 号

Application Number:

特願2002-241877

[ST.10/C]:

[JP 2002-241877]

出 願 人

Applicant(s):

松下電器産業株式会社

2003年 3月14日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田 信一郎



出証番号 出証特2003-3017285

【書類名】 特許願

【整理番号】 5037730129

【提出日】 平成14年 8月22日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/45

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式
社内

 【氏名】 小川 一

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式
社内

 【氏名】 瓶子 岳人

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式
社内

 【氏名】 高山 秀一

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式
社内

 【氏名】 坂田 俊幸

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式
社内

 【氏名】 道本 昌平

【特許出願人】

 【識別番号】 000005821

 【氏名又は名称】 松下電器産業株式会社

【代理人】

【識別番号】 100090446

【弁理士】

【氏名又は名称】 中島 司朗

【手数料の表示】

【予納台帳番号】 014823

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9003742

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 命令スケジューリング方法、命令スケジューリング装置、及びプログラム

【特許請求の範囲】

【請求項 1】 対象プログラム中に記述された複数命令のうち 1 つ以上の命令を対象命令とし、各対象命令に、自命令を処理するハードウェア資源固有の制約条件と、当該ハードウェア資源によって処理される他命令との間に存するデータ依存関係とに基づいて優先順位を算出する優先順位算出ステップと、

最も高い優先順位を算出された対象命令の実行時期を決定する実行時期決定ステップと

を含むことを特徴とする命令スケジューリング方法。

【請求項 2】 前記優先順位算出ステップは、

前記各対象命令に対し、前記データ依存関係にのみ基づいて先行制約順位を算出する先行制約順位算出サブステップと、

前記各対象命令が、自命令と同一の先行制約順位を算出され、かつ前記データ依存関係にある後続命令を持つ場合、自命令とその後続命令とを前記ハードウェア資源が並行して処理可能か否かを判断する資源制約評価サブステップとを含み、

否と判断された各対象命令に対し、前記算出された先行制約順位を繰り上げて得られる優先順位を算出し、その他の各対象命令に対し、前記算出された先行制約順位と等しい優先順位を算出する

ことを特徴とする請求項 1 に記載の命令スケジューリング方法。

【請求項 3】 前記優先順位算出ステップは、

前記各対象命令に対し、（１）前記データ依存関係にある後続命令を持たない場合 1、（２）逆依存関係又は出力依存関係にある 1 つ以上の後続命令を持つ場合、当該後続命令に算出された最大値、及び（３）真の依存関係にある 1 つ以上の後続命令を持つ場合、当該後続命令に算出された最大値よりも 1 大きい値を、それぞれ先行制約順位として算出する先行制約順位算出サブステップと、

前記各対象命令に対し、自命令を処理するハードウェア資源によって処理され

実行時期未決定なる命令の総数を、当該ハードウェア資源が並行して処理可能な最大命令数で除して得られる資源制約値を算出する資源制約評価サブステップとを含み、

当該資源制約値が当該先行制約順位よりも大きい場合、前記対象命令に当該資源制約値を優先順位として算出し、その他の場合、前記対象命令に当該先行制約順位を優先順位として付与し、

前記実行時期決定ステップは、最も大きい優先順位を算出された対象命令の実行時期を決定する

ことを特徴とする請求項 1 に記載の命令スケジューリング方法。

【請求項 4】 対象プログラム中に記述された複数命令の実行時期をそれぞれ決定する命令スケジューリング方法であって、

第 1 命令の実行時期を決定した場合、第 2 命令の実行時期を所定期間に含まれるように決定できなくなるか否かを、当該第 2 命令を処理するハードウェア資源固有の制約条件に基づいて判断する決定可否判断ステップと、

当該決定ができなくなると判断された場合、当該第 1 命令の実行時期の決定を撤回すると共に、当該第 1 命令と異なる命令の実行時期を決定する再決定ステップと

を含むことを特徴とする命令スケジューリング方法。

【請求項 5】 前記所定期間は、クロックサイクル数により示され、前記決定可否判断ステップは、

前記第 2 命令を処理するハードウェア資源によって処理され実行時期未決定なる命令の総数を、当該ハードウェア資源が並行して処理可能な最大命令数で除して得られる資源制約値を算出する資源制約評価サブステップを含み、

当該資源制約値が前記クロックサイクル数よりも大きければ、前記決定ができなくなると判断する

ことを特徴とする請求項 4 に記載の命令スケジューリング方法。

【請求項 6】 入力プログラムを、複数命令によって記述された中間プログラムに変換し、当該中間プログラムに対して請求項 1 乃至 5 の何れかに記載の命令スケジューリング方法を適用することにより、当該複数命令の実行時期を決定

する

ことを特徴とするプログラム変換方法。

【請求項 7】 対象プログラム中に記述された複数命令のうち 1 つ以上の命令を対象命令とし、各対象命令に、自命令を処理するハードウェア資源固有の制約条件と、当該ハードウェア資源によって処理される他命令との間に存するデータ依存関係とに基づいて優先順位を算出する優先順位算出手段と、

最も高い優先順位を算出された対象命令の実行時期を決定する実行時期決定手段と

を備えることを特徴とする命令スケジューリング装置。

【請求項 8】 対象プログラム中に記述された複数命令の実行時期をそれぞれ決定する命令スケジューリング装置であって、

第 1 命令の実行時期を決定した場合、第 2 命令の実行時期を所定期間に含まれるように決定できなくなるか否かを、当該第 2 命令を処理するハードウェア資源固有の制約条件に基づいて判断する決定可否判断手段と、

当該決定ができなくなると判断された場合、当該第 1 命令の実行時期の決定を撤回すると共に、当該第 1 命令と異なる命令の実行時期を決定する再決定手段とを備えることを特徴とする命令スケジューリング装置。

【請求項 9】 命令スケジューリングをコンピュータを用いて行うための、コンピュータ実行可能なプログラムであって、

対象プログラム中に記述された複数命令のうち 1 つ以上の命令を対象命令とし、各対象命令に、自命令を処理するハードウェア資源固有の制約条件と、当該ハードウェア資源によって処理される他命令との間に存するデータ依存関係とに基づいて優先順位を算出する優先順位算出ステップと、

最も高い優先順位を算出された対象命令の実行時期を決定する実行時期決定ステップと

を前記コンピュータに実行させることを特徴とするプログラム。

【請求項 10】 対象プログラム中に記述された複数命令の実行時期を、コンピュータを用いてそれぞれ決定するための、コンピュータ実行可能なプログラムであって、

第 1 命令の実行時期を決定した場合、第 2 命令の実行時期を所定期間に含まれるように決定できなくなるか否かを、当該第 2 命令を処理するハードウェア資源固有の制約条件に基づいて判断する決定可否判断ステップと、

当該決定ができなくなると判断された場合、当該第 1 命令の実行時期の決定を撤回すると共に、当該第 1 命令と異なる命令の実行時期を決定する再決定ステップと

を前記コンピュータに実行させることを特徴とするプログラム。

【請求項 11】 請求項 9 又は請求項 10 のプログラムを記録していることを特徴とする、コンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、命令スケジューリング方法、及び命令スケジューリング装置に関し、特に、命令を処理するハードウェア資源固有の制約条件を考慮して命令をスケジューリングする技術に関する。

【0002】

【従来の技術】

従来の並列処理プロセッサを対象として命令をスケジューリングする命令スケジューリング装置は、入力プログラム中に記述された複数命令それぞれの好適な実行時期を決定する。

このような命令スケジューリング装置は、並列プロセッサを対象とする従来のコンパイラ装置に一般に備えられており、コンパイル後のプログラムに含まれる各命令の好適な実行時期を決定し、当該決定された実行時期に従って各命令を並べ換えることにより、並列最適化された目的プログラムを生成する。

【0003】

従来のある種の命令スケジューリング装置は、リストスケジューリングと呼ばれる方法を用いて命令スケジューリングを行う。当該方法は、入力プログラム中の各命令に命令間に存するデータ依存関係にのみ基づいて優先順位（本願発明に係る優先順位と特に区別する場合、先行制約順位と称する）を算出し、実行時期

未決定なる命令のうち最高の先行制約順位を算出された命令の実行時期を決定する処理を、全ての命令の実行時期が決定されるまで繰り返す。

【 0 0 0 4 】

ここで、データ依存関係とは、同一のハードウェア資源により処理される命令間に存する順序関係を言い、従来、（１）入力プログラムにおいて先に記述された命令（先行命令と称する）が定義した資源を、同じく後に記述された命令（後続命令と称する）が参照する真の依存関係、（２）先行命令が参照した資源を後続命令が定義する逆依存関係、及び（３）先行命令が定義した資源を後続命令がさらに定義する出力依存関係の３種類に分類されている。

【 0 0 0 5 】

これらの関係にある命令間の実行順序を変更すると、プログラムの実行結果の同一性が保証されなくなるため、命令スケジューリング装置は、データ依存関係にある命令間の実行順序が維持されるよう各命令の実行時期を決定する。

図 1 4 は、従来装置が行う命令スケジューリング処理の一例を示すフローチャートである。従来の命令スケジューリング処理は、大きく、データ依存グラフ生成処理 S 9 1 0、優先順位算出処理 S 9 2 0、及び実行時期決定処理 S 9 3 0 に分けて実行される。

（データ依存グラフ生成処理 S 9 1 0）

従来装置は、まず、入力プログラムにおける命令間のデータ依存関係を表すデータ依存グラフを生成する。当該データ依存グラフは、有向非循環グラフであり、入力プログラムの個々の命令に対応してそのノードが設けられ、データ依存関係があると判定された命令に対応するノード間に、先行命令から後続命令に向かうアークが設けられる。

【 0 0 0 6 】

図 1 5 は、従来装置に入力される入力プログラムの一例である。

図 1 6 は、図 1 5 の入力プログラムに対して従来装置が生成するデータ依存グラフである。

（優先順位算出処理 S 9 2 0）

従来装置は、個々の命令に先行制約順位を算出する。算出される先行制約順位

は、一例として、（１）データ依存関係にある後続命令を持たない命令に対して 1、（２）逆依存関係又は出力依存関係にある 1 つ以上の後続命令を持つ命令に対して、当該後続命令に算出された最大値、及び（３）真の依存関係にある 1 つ以上の後続命令を持つ命令に対して、当該後続命令に算出された最大値よりも 1 大きい値であり、数値が大きいほど高い順位を表す。

【 0 0 0 7 】

この先行制約順位は、具体的に、まず前述したデータ依存グラフにおいて真の依存関係、逆依存関係、及び出力依存関係を表すアークに、それぞれ、1、0、及び 0 なる重み値を付与し、次に各ノードからデータ依存関係にある後続命令を終端ノードまで辿るパス上のアークに付与された重み値を累計し、さらに 1 を加えることによって算出される。終端ノードに至るパスが複数あり、複数の値が算出される場合、その最大値を採用する。

【 0 0 0 8 】

図 1 6 において、算出される先行制約順位を各ノードに付記する。

先行制約順位の値は、真の依存関係、逆依存関係、及び出力依存関係にある命令間のレイテンシを、それぞれ一律に 1、0、及び 0 とし、かつ、データ依存関係にある命令間の実行順序を維持する限りにおいて、各パス上の先頭命令を含む実行時期未決定なる命令を全て実行するための最小所要時間に相当する。

【 0 0 0 9 】

当該値が最大のパスは、クリティカルパスと呼ばれる。従来装置は、クリティカルパスの先頭命令（即ち、最大の優先順位を算出された命令）をできるだけ早い時期に実行させるようにスケジューリングすることにより、全命令の実行所要時間を極小化する。

（実行時期決定処理 S 9 3 0）

従来装置は、データ依存関係にある命令の実行順序を維持するために、データ依存関係にある先行命令が存在しないか、又は存在するならば全て実行時期決定済である命令のみを配置可能と判断する。

【 0 0 1 0 】

そして、各命令の配置可否判断、配置可能と判断された命令のうち最も高い先

行制約順位を算出された命令の選択、及び、当該選択された命令の実行時期決定処理を、全ての命令の実行時期が決定されるまで繰り返す。

命令の実行時期は、具体的には、当該命令を実行すべきクロックサイクルとして決定される。命令の実行時期をあるクロックサイクルに決定することを、命令をそのクロックサイクルに配置するとも言う。

【 0 0 1 1 】

従来装置は、命令を、（１）逆依存関係又は出力依存関係にある先行命令が配置されたクロックサイクルと同じかそれ以降、かつ真の依存関係にある先行命令が配置されたクロックサイクルよりも後で、（２）ハードウェア資源が当該命令を処理可能な最先のクロックサイクルに配置する。

従来装置は、これにより、プログラムの実行結果の同一性を保つと共に、全命令を極小のクロックサイクルに配置する。

【 0 0 1 2 】

図 1 7 は、対象プロセッサに備えられた命令デコーダ、演算部、及びメモリアクセス部が、１つのクロックサイクルあたり、それぞれ 2 命令、2 命令、及び 1 命令を並行して処理可能である場合に、図 1 5 のプログラムの各命令が配置されるクロックサイクルを示している。同図中、クロックサイクル欄 9 0 1 は、クロックサイクルを相対番号で示し、命令 1 欄 9 0 2 及び命令 2 欄 9 0 3 は、各クロックサイクルに配置された命令と、それが何回目の決定によって配置されたかを示している。

【 0 0 1 3 】

メモリアクセス部により処理される命令 E、命令 F、及び命令 G は、一クロックサイクルあたり一命令のみを実行可能である。そのため、これらの命令はサイクル 3、サイクル 4、及びサイクル 5 に配置され、その結果、サイクル 4 には、命令 F のみが配置される。

従来のコンパイラは、このようにして配置された命令をクロックサイクルの早い順に並べると共に、各クロックサイクルの末尾の命令に、クロックサイクルの境界を示す境界情報を付加することにより、並列最適化された目的プログラムを生成する。当該境界情報は、例えば 1 ビットのフラグ情報として表される。対象

プロセッサは、当該境界情報を付加された命令とその次の命令とを、それぞれ異なるクロックサイクルにおいて実行する。

【 0 0 1 4 】

図 1 7 の例について言えば、各命令は再び図 1 5 に示した順序で出力されると共に、命令 A、命令 C、命令 E、命令 F、及び命令 G に当該境界情報が付加される。

並列最適化された目的プログラムは、対象プロセッサにおいて、並列最適化されないプログラムに比べて、より少ないクロックサイクルで実行されることが期待される。

【 0 0 1 5 】

【発明が解決しようとする課題】

しかしながら、上記従来の命令スケジューリング技術によれば、全命令が必ずしも最小のクロックサイクルに配置されない、即ち並列最適化が十分に行われない場合があるという問題がある。

例えば、図 1 5 の例について言えば、第 2 決定において命令 E を選択してサイクル 2 に配置すれば、命令 F、及び命令 G が、サイクル 3、及びサイクル 4 にそれぞれ配置できるとともに、命令 B、命令 C、及び命令 D もまた、サイクル 2、サイクル 3、及びサイクル 4 にそれぞれ配置でき、全命令が 4 クロックサイクルに配置されることとなる（図 5 参照）。

【 0 0 1 6 】

しかしながら、従来技術によれば、先行制約順位をデータ依存関係にのみ基づいて算出するので、第 2 決定において命令 E が選択されるような先行制約順位は算出され得ず、このような最適な配置は行われない。

上記の問題に鑑み、本発明は、従来よりも少ないクロックサイクルに全命令を配置できる命令スケジューリング方法、及び命令スケジューリング装置の提供を目的とする。

【 0 0 1 7 】

【課題を解決するための手段】

上記問題を解決するため、本発明の命令スケジューリング方法は、対象プログ

ラム中に記述された複数命令のうち1つ以上の命令を対象命令とし、各対象命令に、自命令を処理するハードウェア資源固有の制約条件と、当該ハードウェア資源によって処理される他命令との間に存するデータ依存関係とに基づいて優先順位を算出する優先順位算出ステップと、最も高い優先順位を算出された対象命令の実行時期を決定する実行時期決定ステップとを含む。

【0018】

また、本発明の命令スケジューリング方法は、対象プログラム中に記述された複数命令の実行時期をそれぞれ決定する命令スケジューリング方法であって、第1命令の実行時期を決定した場合、第2命令の実行時期を所定期間に含まれるように決定できなくなるか否かを、当該第2命令を処理するハードウェア資源固有の制約条件に基づいて判断する決定可否判断ステップと、当該決定ができなくなると判断された場合、当該第1命令の実行時期の決定を撤回すると共に、当該第1命令と異なる命令の実行時期を決定する再決定ステップとを含んでもよい。

【0019】

また、本発明の命令スケジューリング装置は、対象プログラム中に記述された複数命令のうち1つ以上の命令を対象命令とし、各対象命令に、自命令を処理するハードウェア資源固有の制約条件と、当該ハードウェア資源によって処理される他命令との間に存するデータ依存関係とに基づいて優先順位を算出する優先順位算出手段と、最も高い優先順位を算出された対象命令の実行時期を決定する実行時期決定手段とを備える。

【0020】

また、本発明の命令スケジューリング装置は、対象プログラム中に記述された複数命令の実行時期をそれぞれ決定する命令スケジューリング装置であって、第1命令の実行時期を決定した場合、第2命令の実行時期を所定期間に含まれるように決定できなくなるか否かを、当該第2命令を処理するハードウェア資源固有の制約条件に基づいて判断する決定可否判断手段と、当該決定ができなくなると判断された場合、当該第1命令の実行時期の決定を撤回すると共に、当該第1命令と異なる命令の実行時期を決定する再決定手段とを備えてもよい。

【0021】

【発明の実施の形態】

(第 1 の実施の形態)

第 1 の実施の形態における命令スケジューリング装置は、入力プログラム中に記述された複数命令の各々に、データ依存関係と、ハードウェア資源固有の制約条件とに基づいて優先順位を算出し、算出された優先順位を用いて命令の選択と配置とを行う。

【0 0 2 2】

より詳細には、データ依存関係にあって、かつ同一の先行制約順位を算出される 2 つの命令について、対象プロセッサにおけるハードウェア資源が両者を並行して処理可能か否かを判断する。そして、否と判断された場合その先行命令に対して当該先行制約順位を繰上げて得られる順位を本発明に係る優先順位とし、それ以外の命令に対して当該先行制約順位を本実施の形態に係る優先順位とする。そして、最も高い優先順位を算出された未配置命令の選択と配置とを、全ての命令が配置されるまで繰り返す。

【0 0 2 3】

本命令スケジューリング装置は、言うなれば、先行制約順位としては同一値を算出されるが、ハードウェア資源に関する制約のために並行して処理されない 2 つの命令について、その先行命令の優先順位を従来の先行制約順位よりも高く算出することで、資源制約のために生じるクリティカルパスを発見する。従来装置は、資源制約のために生じるクリティカルパスを発見していなかった。

【0 0 2 4】

本命令スケジューリング装置は、そのようなクリティカルパス上の先頭命令をできるだけ早いクロックサイクルに配置するので、資源制約のために並行処理されない命令を含む全命令を、従来よりも少ないクロックサイクルに配置できる。

(全体構成)

図 1 は、第 1 の実施の形態におけるコンパイラ装置 1 0 0 の全体構成を示す機能ブロック図である。コンパイラ装置 1 0 0 は、命令スケジューリング部 1 3 0 として、第 1 の実施の形態に係る命令スケジューリング装置を含んでいる。

【0 0 2 5】

コンパイラ装置 1 0 0 は、ソースファイル 1 0 1 からソースプログラムを取得し、当該ソースプログラムをコンパイルして得られたプログラムから、並列最適化された目的プログラムを生成し、当該目的プログラムを目的ファイル 1 0 2 へ出力する。

コンパイラ装置 1 0 0 は、コンパイラ上流部 1 1 0、アセンブラコード生成部 1 2 0、命令スケジューリング部 1 3 0、及び出力部 1 7 0 から構成される。命令スケジューリング部 1 3 0 は、データ依存関係解析部 1 4 0、優先順位算出部 1 5 0、及び実行時期決定部 1 6 0 から構成される。優先順位算出部 1 5 0 は、先行制約順位算出部 1 5 1、及び資源制約評価部 1 5 2 を含み、実行時期決定部 1 6 0 は、命令選択部 1 6 1 を含む。

【 0 0 2 6 】

コンパイラ装置 1 0 0 は、具体的にはプロセッサ、プログラムを記憶している ROM (Read Only Memory)、作業用の RAM (Random Access Memory)、ディスク装置等のソフトウェア及びハードウェアにより実現される。前記各部の機能は、プロセッサが ROM に記憶されているプログラムを実行することにより実現される。前記各部の間におけるデータの受け渡しは、RAM、ディスク装置等のハードウェアを介して行われる。

【 0 0 2 7 】

コンパイラ上流部 1 1 0 は、ソースファイル 1 0 1 からソースプログラムを読み出し、字句解析処理及び構文解析処理を行うことにより、中間コード列を生成する。

アセンブラコード生成部 1 2 0 は、コンパイラ上流部 1 1 0 により生成された中間コード列から、アセンブラコード列を生成する。

【 0 0 2 8 】

命令スケジューリング部 1 3 0 は、アセンブラコード生成部 1 2 0 により生成されたアセンブラコード列に含まれる各命令に、他の命令との間のデータ依存関係と、自命令を処理するハードウェア資源固有の制約条件とに基づいて優先順位を算出する。そして、実行時期未決定なる命令のうち最も高い優先順位を算出された命令を選択し、選択された命令の実行時期を決定する処理を、全ての命令の

実行時期が決定されるまで繰り返す。命令スケジューリング部 1 3 0 の詳細について、後述する。

【 0 0 2 9 】

出力部 1 7 0 は、各命令を、配置されたクロックサイクルの早い順に、従来技術の項で述べた境界情報と共に出力する。

以下、コンパイラ装置 1 0 0 が対象とするプロセッサの構成について説明し、その後、命令スケジューリング部 1 3 0 の詳細を説明する。

(対象プロセッサ)

図 2 は、コンパイラ装置 1 0 0 が対象とするプロセッサの一構成例を示す機能ブロック図である。本図は、本願発明に言うハードウェア資源固有の制約条件の一具体例を説明するために掲げられ、当該説明に必要な部分のみを簡略に示している。

【 0 0 3 0 】

プロセッサ 8 0 0 は、大きく分けて、命令供給部 8 1 0、解読部 8 2 0、実行部 8 3 0 から構成される。

命令供給部 8 1 0 は、I A (Instruction Address) バス、及び I D (Instruction data) バスを介して、命令フェッチ部 8 1 1 により、図示しない外部メモリから命令をフェッチし、フェッチされた命令を第 1 命令レジスタ 8 1 2 及び第 2 命令レジスタ 8 1 3 に保持すると共に、1つのクロックサイクルに 2 命令を並行して解読部 8 2 0 へ供給する。

【 0 0 3 1 】

解読部 8 2 0 は、第 1 命令デコーダ 8 2 1 及び第 2 命令デコーダ 8 2 2 を含み、1つのクロックサイクルに 2 命令を並行して解読し、その解読結果に応じた制御信号を実行部 8 3 0 へ供給する。

実行部 8 3 0 は、解読部 8 2 0 から供給された制御信号に応じて動作する。第 1 演算部 8 3 1 と第 2 演算部 8 3 2 とは、それぞれ専用のバスラインでレジスタファイル 8 3 3、及び条件フラグレジスタ 8 3 4 と接続され、1つのクロックサイクルに 2 つの命令に関する演算処理を並行して行う。メモリアクセス部 8 3 5 は、O A (Operand Address) バス、及び O D (Operand data) バスを介して、

1つのクロックサイクルに1つの命令に関するメモリアクセス処理を行う。

【 0 0 3 2 】

プロセッサ 8 0 0 は、このような構成を有するため、1つのクロックサイクルに、演算部を用いて処理される命令であれば最大2つ、メモリアクセスユニットを用いて処理される命令であれば最大1つを処理可能である。これが、プロセッサ 8 0 0 の資源制約条件となる。

(命令スケジューリング部 1 3 0 の詳細)

第1の実施の形態における命令スケジューリング部 1 3 0 の詳細についてフローチャートを参照しながら説明する。

【 0 0 3 3 】

図3は、第1の実施の形態における命令スケジューリング処理を示すフローチャートである。

(ステップ S 1 0 1) データ依存関係解析部 1 4 0 は、アセンブラコード生成部 1 2 0 により生成されたアセンブラコード列に含まれる命令間のデータ依存関係を表すデータ依存グラフを、従来技術と同様にして生成する。

【 0 0 3 4 】

(ステップ S 1 0 2) 先行制約順位算出部 1 5 1 は、データ依存関係解析部 1 4 0 により生成されたデータ依存グラフにおいて、真の依存関係、逆依存関係、及び出力依存関係を表す各アークに、それぞれ1、0、及び0なる重み値を付与する。この処理は、従来技術において先行制約順位を算出する処理に相当する。

(ステップ S 1 0 3) 重み値0を付与された各アークについて、ステップ S 1 0 6 までを実行する。

【 0 0 3 5 】

(ステップ S 1 0 4) 資源制約評価部 1 5 2 は、当該アークの両端の命令、即ち、先行制約値として同一値を算出される命令について、ハードウェア資源が当該命令を並行して処理可能か否か判断する。否と判断された場合、ステップ S 1 0 5 を実行する。

(ステップ S 1 0 5) 当該アークの重み値を1に変更する。

【 0 0 3 6 】

(ステップ S 1 0 6) ステップ S 1 0 3 から繰り返す。

(ステップ S 1 0 7) 優先順位算出部 1 5 0 は、各ノードからデータ依存関係にある後続命令を終端ノードまで辿るパス上のアークに付与され、資源制約評価部 1 5 2 によって変更された後の重み値を累計し、さらに 1 を加えることにより、ノード毎に優先順位を算出する。ステップ S 1 0 5 において変更した重み値を累計することで、先行制約順位を繰上げて得られる優先順位が算出される。

【 0 0 3 7 】

(ステップ S 1 0 8) 実行時期未決定なる命令がある間、ステップ S 1 1 1 まですてを実行する。

(ステップ S 1 0 9) 命令選択部 1 6 1 は、実行時期未決定なる命令に対応するノードのうち、優先順位算出部 1 5 0 により最も高い優先順位を算出されたノードを選択する。

【 0 0 3 8 】

(ステップ S 1 1 0) 実行時期決定部 1 6 0 は、命令選択部 1 6 1 により選択されたノードに対応する命令を、(1) 命令と逆依存関係又は出力依存関係にある先行命令が配置されたクロックサイクルと同じかそれよりも後、かつ当該命令と真の依存関係にある先行命令が配置されたクロックサイクルよりも後で、(2) ハードウェア資源が当該命令を処理可能な最も早いクロックサイクルに当該命令を配置する。

【 0 0 3 9 】

(ステップ S 1 1 1) ステップ S 1 0 8 から繰り返す。

(具体例)

図 4 は、図 1 5 のプログラムから、データ依存関係解析部 1 4 0 によって生成され、先行制約順位算出部 1 5 1 によって各アークに重み付けされたデータ依存グラフを示している。カッコ内の数値は、各アークに付与された重み値を示す。

【 0 0 4 0 】

資源制約評価部 1 5 2 は、重み値 0 を付与されたアークの両端の命令、即ち命令 E 及び命令 F、並びに命令 F 及び命令 G が、何れもメモリアクセスユニットを用いて処理される命令であることから、同一のクロックサイクルにおいて並行し

て処理不可能であると判断し、重み値をそれぞれ 1 に変更する。図 4 において、 $(0 \rightarrow 1)$ なる表記は重み値の変更を示す。

【 0 0 4 1 】

優先順位算出部 1 5 0 は、当該変更後の重み値を累計することにより優先順位を算出する。図 4 において、各ノードに付記した数値は、当該算出された優先順位を示す。ノード A の優先順位は、パス A E F G から算出された 4 が採られる。

図 5 は、図 4 のデータ依存グラフから算出された優先順位を用いて行われる命令の配置を示している。表記法は図 1 7 と同様である。命令 E の優先順位が 3 と算出され、第 2 決定で命令 E がサイクル 2 に配置されることによって、図 1 7 の配置よりも 1 クロックサイクル少ない 4 クロックサイクルに全ての命令が配置される。

(第 1 のまとめ)

以上述べたように、第 1 の実施の形態における命令スケジューリング装置は、データ依存関係にあつて、かつ同一の先行制約順位を算出される 2 つの命令が、対象プロセッサにおけるハードウェア資源によって並行処理されない場合、その先行命令の優先順位を従来の先行制約順位よりも高く算出する。

【 0 0 4 2 】

これにより、従来の技術では発見されなかった資源制約のために生じるクリティカルパスを発見し、そのようなクリティカルパス上の先頭命令をできるだけ早いクロックサイクルに配置するので、資源制約のために並行処理されない命令を含む全命令を、従来よりも少ないクロックサイクルに配置できる。

(第 2 の実施の形態)

第 2 の実施の形態における命令スケジューリング装置は、まず、入力プログラム中に記述された複数命令の各々に先行制約順位を算出する。そして、配置可能な命令の各々に、自命令を処理するハードウェア資源によって処理される未配置命令の総数を、当該ハードウェア資源が並行して処理可能な最大命令数で除して得られる資源制約値を算出し、当該資源制約値及び前記先行制約順位の何れか大きい方を優先順位とし、最も高い優先順位を持つ命令の選択と配置とを、全ての命令が配置されるまで繰り返す。

【 0 0 4 3 】

資源制約値は、当該ハードウェア資源によって処理される未配置命令を全て実行するための最小所要時間に相当する。

本命令スケジューリング装置は、この資源制約値を算出する点、及び、命令が配置されるたびに優先順位を算出し直す点で、第 1 の実施の形態と異なる。

以下、第 1 の実施の形態と同一の事項については説明を省略し、異なる事項について主に説明する。

（全体構成）

第 2 の実施の形態におけるコンパイラ装置は、第 1 の実施の形態のコンパイラ装置と同一の全体構成（図 1 参照）を有している。命令スケジューリング部 1 3 0 として、第 2 の実施の形態に係る命令スケジューリング装置を含み、その詳細な動作が、第 1 の実施の形態と異なる。

（命令スケジューリング部 1 3 0 の詳細）

第 2 の実施の形態における命令スケジューリング部 1 3 0 の詳細について、フローチャートを参照しながら説明する。

【 0 0 4 4 】

図 6 は、第 2 の実施の形態における命令スケジューリング処理を示すフローチャートである。

（ステップ S 2 0 1）データ依存関係解析部 1 4 0 は、アセンブラコード生成部 1 2 0 により生成されたアセンブラコード列に含まれる命令間のデータ依存関係を表すデータ依存グラフを生成する。

【 0 0 4 5 】

（ステップ S 2 0 2）先行制約順位算出部 1 5 1 は、データ依存関係解析部 1 4 0 により生成されたデータ依存グラフにおいて、真の依存関係、逆依存関係、及び出力依存関係を表す各アークに、それぞれ、1、0、及び 0 なる重み値を付与し累計することにより、先行制約順位を算出する。

（ステップ S 2 0 3）未配置命令がある間ステップ S 2 1 3 までを実行する。

【 0 0 4 6 】

（ステップ S 2 0 4）命令スケジューリング部 1 3 0 は、データ依存関係にあ

る先行命令を持たないか、持つならば全て配置済となっている命令を配置可能命令と認識して、そのリストを作成する。

(ステップ S 2 0 5) 作成されたリスト中の各命令についてステップ S 2 1 0 までを実行する。

【 0 0 4 7 】

(ステップ S 2 0 6) 資源制約評価部 1 5 2 は、当該命令に対し、自命令を処理するハードウェア資源によって処理される未配置命令の数を、当該ハードウェア資源が並行して処理可能な最大命令数で除して得られる資源制約値を算出する。

(ステップ S 2 0 7) 算出された資源制約値が、当該命令に算出された先行制約順位よりも大きい場合、ステップ S 2 0 8 を実行し、その他の場合、ステップ S 2 0 9 を実行する。

【 0 0 4 8 】

(ステップ S 2 0 8) 当該資源制約値を、当該命令の優先順位とする。

(ステップ S 2 0 9) 当該先行制約順位を、当該命令の優先順位とする。

(ステップ S 2 1 0) ステップ S 2 0 5 から繰り返す。

(ステップ S 2 1 1) 命令選択部 1 6 1 は、実行時期未決定なる命令に対応するノードのうち、優先順位算出部 1 5 0 により最も高い優先順位を算出されたノードを選択する。

【 0 0 4 9 】

(ステップ S 2 1 2) 実行時期決定部 1 6 0 は、命令選択部 1 6 1 により選択されたノードに対応する命令を、(1) 命令と逆依存関係又は出力依存関係にある先行命令が配置されたクロックサイクルと同じかそれよりも後、かつ当該命令と真の依存関係にある先行命令が配置されたクロックサイクルよりも後で、(2) ハードウェア資源が当該命令を処理可能な最も早いクロックサイクルに当該命令を配置する。

【 0 0 5 0 】

(ステップ S 2 1 3) ステップ S 2 0 3 から繰り返す。

(具体例)

再び、図 1 5 のプログラムを例に説明する。第 2 の実施の形態におけるデータ依存関係解析部 1 4 0 は、図 1 6 に示した従来と同様のデータ依存グラフを生成し、先行制約順位算出部 1 5 1 は、当該データ依存グラフから従来の優先順位である先行制約順位を算出する。

【 0 0 5 1 】

図 7 及び図 8 は、命令スケジューリング部 1 3 0 による各命令の配置過程を示している。

両図中、命令欄 3 0 1 は各命令を符号で示し、使用資源欄 3 0 2 は当該命令の処理に使用される資源がメモリアクセス部、又は演算部であることを、それぞれ記号 M、A で示し、先行制約順位欄 3 0 3 は、当該命令に算出される先行制約順位を示している。

【 0 0 5 2 】

第 1 決定欄 3 1 0 乃至第 7 決定欄 3 7 0 は、配置の決定順に、それぞれの決定について、決定前の配置状態、算出される資源制約値、算出される優先順位を示している。配置状態について、未配置でかつその回に配置不可、未配置でかつその回に配置可能、及び配置済である状態が、未決定、配置可能、及び決定されたサイクル番号により、それぞれ表される。

【 0 0 5 3 】

配置結果欄 3 8 0 は、最終的に各命令が配置されるサイクル番号を示している。

以下、各回の決定について、詳細に説明する。

（第 1 決定）命令スケジューリング部 1 3 0 は、データ依存関係にある先行命令を持たない命令 A を唯一の配置可能命令と認識し、A のみを要素とするリスト {A} を作成する。

【 0 0 5 4 】

資源制約評価部 1 5 2 は、命令 A に対し、資源制約値を算出する。命令 A は、メモリアクセス部によって処理される命令である。この時点で実行時期未決定なる命令のうちメモリアクセス部によって処理される命令は、命令 A、命令 E、命令 F、及び命令 G の 4 つであり、その数をメモリアクセス部が並行して処理可能

な最大命令数 1 で除して得られる 4 を命令 A の資源制約値とする。

【 0 0 5 5 】

当該資源制約値は、先行制約順位よりも大きいため、当該資源制約値 4 が命令 A の優先順位とされる。

命令選択部 1 6 1 は、命令 A を選択し、実行時期決定部 1 6 0 は、命令 A をサイクル 1 に配置する。

(第 2 決定) 命令 A が配置済となったことで、優先順位算出部 1 5 0 は命令 B、命令 C、及び命令 E を配置可能命令と認識し、リスト {A、C、E} を作成する。

【 0 0 5 6 】

資源制約評価部 1 5 2 は、命令 B に対し、資源制約値を算出する。命令 B は、演算部によって処理される命令である。この時点で演算部によって処理される未配置命令は、命令 B、命令 C、及び命令 D の 3 つであり、その数を演算部が並行して処理可能な最大命令数 2 で除して得られる 1. 5 を命令 B の資源制約値とする。

【 0 0 5 7 】

当該資源制約値は、先行制約順位よりも大きくないため、先行制約順位 2 が命令 B の優先順位とされる。

資源制約評価部 1 5 2 は、命令 C の優先順位を、命令 B と同様にして 2 と算出する。

資源制約評価部 1 5 2 は、命令 E に対し、資源制約値を算出する。命令 E は、メモリアクセス部によって処理される命令である。この時点でメモリアクセス部によって処理される未配置命令は、命令 E、命令 F、及び命令 G の 3 つであり、その数をメモリアクセス部が並行して処理可能な最大命令数 1 で除して得られる 3 を命令 E の資源制約値とする。

【 0 0 5 8 】

当該資源制約値は、先行制約順位よりも大きいため、優先順位算出部 1 5 0 は、当該資源制約値 3 を命令 E の優先順位とする。

命令選択部 1 6 1 は、最大の優先順位を算出された命令 E を選択し、実行時期

決定部 1 6 0 は、命令 E を、命令 A が配置されたサイクルの次サイクル以降、最
先のサイクル 2 に配置する。

【 0 0 5 9 】

(第 3 決定) 命令 A 及び命令 E が配置済となったことで、優先順位算出部 1 5
0 は、それらを先行命令とする命令 B、命令 C、及び命令 F を配置可能命令と認
識し、リスト {A、C、F} を作成する。

資源制約評価部 1 5 2 は、命令 C、及び命令 B に対し、第 2 決定と同様にして
優先順位 2 を算出する。

【 0 0 6 0 】

資源制約評価部 1 5 2 は、命令 F に対し資源制約値 2 を算出する。当該資源制
約値は先行制約順位よりも大きいため、優先順位算出部 1 5 0 は当該資源制約値
2 を命令 F の優先順位とする。

命令 B、命令 C、及び命令 F に同一の優先順位が算出されるため、命令選択部
1 6 1 は、元のプログラムにおける記述順に従って命令 B を選択する。実行時期
決定部 1 6 0 は、命令 A が配置されたサイクル 1 の次サイクル以降、最先のサイ
クルに配置する。対象プロセッサにおいて、命令 B と、第 2 決定でサイクル 2 に
配置された命令 E とが、各構成の動作並列度数の最大値を超えることなく並行し
て処理可能であると判断されることから、この最先のサイクルはサイクル 2 と決
定される。

【 0 0 6 1 】

(第 4 決定) 以下、簡略に説明する。優先順位算出部 1 5 0 は、配置可能命令
のリスト {C、F} を作成する。資源制約評価部 1 5 2 は、命令 C、及び命令 F
にそれぞれ資源制約値 1、及び 2 を算出し、優先順位算出部 1 5 0 は、命令 C、
及び命令 F にそれぞれ優先順位 2 を算出する。

命令選択部 1 6 1 は、元のプログラムにおける記述順に従って命令 C を選択し
、実行時期決定部 1 6 0 は、命令 C をサイクル 3 に配置する。

【 0 0 6 2 】

(第 5 決定) 優先順位算出部 1 5 0 は、配置可能命令のリスト {D、F} を作
成する。資源制約評価部 1 5 2 は、命令 D、及び命令 F にそれぞれ資源制約値 0

． 5、及び 2 を算出し、優先順位算出部 1 5 0 は、命令 D、及び命令 F にそれぞれ優先順位 1、及び 2 を算出する。

命令選択部 1 6 1 は、命令 F を選択し、実行時期決定部 1 6 0 は、命令 F をサイクル 3 に配置する。

【 0 0 6 3 】

（第 6 決定）優先順位算出部 1 5 0 は、配置可能命令のリスト {D、G} を作成する。資源制約評価部 1 5 2 は、命令 D、及び命令 G にそれぞれ資源制約値 0． 5、及び 1 を算出し、優先順位算出部 1 5 0 は、命令 D、及び命令 G にそれぞれ優先順位 1 を算出する。

命令選択部 1 6 1 は、元のプログラムにおける記述順に従って命令 D を選択し、実行時期決定部 1 6 0 は、命令 D をサイクル 4 に配置する。

【 0 0 6 4 】

（第 7 決定）優先順位算出部 1 5 0 は、配置可能命令のリスト {G} を作成する。優先順位算出部 1 5 0 は、命令 G に優先順位 1 を算出する。

命令選択部 1 6 1 は、命令 G を選択し、実行時期決定部 1 6 0 は、命令 G をサイクル 4 に配置する。

このような配置過程から、第 1 の実施の形態と同様の命令の配置（図 5 参照）が決定される。

（第 2 のまとめ）

以上述べたように、第 2 の実施の形態における命令スケジューリング装置は、配置可能な命令の各々に、資源制約値と先行制約順位の何れか大きい方を優先順位として算出し、最も高い優先順位を算出された命令の選択と配置とを、全ての命令が配置されるまで繰り返す。

【 0 0 6 5 】

これにより、資源制約の厳しい命令を従来よりも早いクロックサイクルに配置するので、そのような命令を含む全命令を従来よりも少ないクロックサイクルに配置できる。

特に、並列処理可能な命令数の小さな資源によって処理される多数の未配置命令があって、それらがデータ依存関係にない場合に、そのような命令に大きな資

源制約値が算出されるので、早いクロックサイクルに適切に配置されるという、特有の効果が生じる。第 1 の実施の形態の命令スケジューリング装置は、データ依存関係にある命令間においてのみ、資源制約に応じて先行命令の優先順位を繰上げるので、そのような場合を適切に処理できない。

（第 3 の実施の形態）

第 3 の実施の形態における命令スケジューリング装置は、まず、入力プログラム中に記述された複数命令の各々に対し、先行制約順位を算出する。

【 0 0 6 6 】

そして、以下に述べる処理を繰り返すことにより、所定目標数のクロックサイクルに命令の配置を試みる。

配置可能命令のうち最大の先行制約順位を算出された命令を配置し、その後、各未配置命令について、当該命令を配置可能なクロックサイクルの残数と、資源制約値とを算出する。当該資源制約値と当該残数との大小関係に基づいて、前記目標数のクロックサイクルに全未配置命令が配置可能か否かを判断する。

【 0 0 6 7 】

否と判断された場合、直前に行った命令の配置を撤回すると共に、その命令を配置可能命令から除外し、他の配置可能命令の配置を試みる。

本命令スケジューリング装置は、資源制約値を用いて未配置命令の配置可否を判断し、否と判断された場合、直前の配置を撤回して他の命令の配置を試みる点で、第 2 の実施の形態と異なる。

【 0 0 6 8 】

以下、第 2 の実施の形態と同一の事項については説明を省略し、異なる事項について主に説明する。

（全体構成）

図 9 は、第 3 の実施の形態におけるコンパイラ装置 4 0 0 の全体構成を示す機能ブロック図である。コンパイラ装置 4 0 0 は、命令スケジューリング部 4 3 0 として、第 3 の実施の形態に係る命令スケジューリング装置を含んでいる。

【 0 0 6 9 】

コンパイラ装置 4 0 0 は、コンパイラ装置 1 0 0 と同様、ソースファイル 1 0

1 に保持されているソースプログラムから、並列最適化された目的プログラムを生成して目的ファイル 1 0 2 へ出力する。

コンパイラ装置 4 0 0 は、第 1 の実施の形態におけるコンパイラ装置 1 0 0 と共通の構成要素を含んでいる。同図中、そのような共通の構成要素を、図 1 に付した符号と同一の符号を用いて示す。

【 0 0 7 0 】

コンパイラ装置 4 0 0 は、コンパイラ上流部 1 1 0、アセンブラコード生成部 1 2 0、命令スケジューリング部 4 3 0、及び出力部 1 7 0 から構成される。命令スケジューリング部 4 3 0 は、データ依存関係解析部 1 4 0、先行制約順位算出部 1 5 1、及び実行時期決定部 4 6 0 から構成される。実行時期決定部 4 6 0 は、命令選択部 1 6 1、決定可否判断部 4 6 2、及び再決定制御部 4 6 4 を含む。決定可否判断部 4 6 2 は、資源制約評価部 1 5 2 を含む。

【 0 0 7 1 】

コンパイラ装置 4 0 0 は、具体的にはプロセッサ、プログラムを記憶している ROM (Read Only Memory)、作業用の RAM (Random Access Memory)、ディスク装置等のソフトウェア及びハードウェアにより実現される。前記各部の機能は、プロセッサが ROM に記憶されているプログラムを実行することにより実現される。前記各部の間におけるデータの受け渡しは、RAM、ディスク装置等のハードウェアを介して行われる。

【 0 0 7 2 】

コンパイラ上流部 1 1 0、アセンブラコード生成部 1 2 0、及び出力部 1 7 0 は、第 1 の実施の形態と共通のため説明を省略し、以下、命令スケジューリング部 4 3 0 について説明する。

(命令スケジューリング部 4 3 0 の詳細)

第 3 の実施の形態における命令スケジューリング部 4 3 0 の詳細についてフローチャートを参照しながら説明する。

【 0 0 7 3 】

図 1 0 は、第 3 の実施の形態における命令スケジューリング処理を示すフローチャートである。

(ステップ S 4 0 1) データ依存関係解析部 1 4 0 は、アセンブラコード生成部 1 2 0 により生成されたアセンブラコード列に含まれる命令間のデータ依存関係を表すデータ依存グラフを生成する。

【 0 0 7 4 】

(ステップ S 4 0 2) 先行制約順位算出部 1 5 1 は、データ依存関係解析部 1 4 0 により生成されたデータ依存グラフにおいて、真の依存関係、逆依存関係、及び出力依存関係を表す各アークに、それぞれ、1、0、及び0なる重み値を付与し累計することにより、先行制約順位を算出する。

(ステップ S 4 0 3) 未配置命令がある間ステップ S 4 1 4 までを実行する。

【 0 0 7 5 】

(ステップ S 4 0 4) 命令スケジューリング部 4 3 0 は、データ依存関係にある先行命令を持たないか、持つならば全て配置済となっている命令を配置可能命令と認識して、そのリストを作成する。

(ステップ S 4 0 5) 命令選択部 1 6 1 は、当該リストから、最大の先行制約順位を算出された命令を選択し、実行時期決定部 4 6 0 は、(1) 命令と逆依存関係又は出力依存関係にある先行命令が配置されたクロックサイクルと同じかそれよりも後、かつ当該命令と真の依存関係にある先行命令が配置されたクロックサイクルよりも後で、(2) ハードウェア資源が当該命令を処理可能な最も早いクロックサイクルに配置する。

【 0 0 7 6 】

(ステップ S 4 0 6) 命令スケジューリング部 4 3 0 は、当該命令を当該リストから除外する。

(ステップ S 4 0 7) ステップ S 4 0 5 において行われた決定によって新たに配置可能となった命令を含む各配置可能命令について、ステップ S 4 1 1 までを実行する。

【 0 0 7 7 】

(ステップ S 4 0 8) 資源制約評価部 1 5 2 は、当該命令を処理するハードウェア資源によって処理される未配置命令の数を、当該ハードウェア資源が並行して処理可能な最大命令数で除して得られる資源制約値を算出する。

決定可否判断部 4 6 2 は、当該命令を配置する余地があるクロックサイクルの数を残サイクル数として算出する。この算出には、対象プロセッサにおいて全ての命令を共通して処理する資源（例えば命令デコーダ）が、1クロックサイクルあたり並行して処理できる命令の最大数（以降、共通最大数と称する）が用いられる。前述のプロセッサ 8 0 0 の例では、この共通最大数は 2 である。

【 0 0 7 8 】

当該残サイクル数は、具体的に、（１）所定目標数のクロックサイクルの中で、（２）当該命令と逆依存関係又は出力依存関係にある先行命令が配置されたクロックサイクルと同じかそれよりも後、かつ当該命令と真の依存関係にある先行命令が配置されたクロックサイクルよりも後において、（３）配置済の命令数が共通最大数に達していないクロックサイクルの数として求められる。

【 0 0 7 9 】

（ステップ S 4 0 9）当該資源制約値が当該残サイクル数を上回っていれば、ステップ S 4 1 0 へ進み、そうでなければ、ステップ S 4 1 3 へ進む。

（ステップ S 4 1 0）当該リストが空なら、ステップ S 4 1 2 へ進み、そうでなければ、ステップ S 4 1 1 へ進む。

（ステップ S 4 1 1）再決定制御部 4 6 4 は、ステップ S 4 0 5 において行われた決定を撤回する。ステップ S 4 0 5 へ戻って他の命令の配置を試みる。

【 0 0 8 0 】

（ステップ S 4 1 2）全命令を所定目標数のクロックサイクルに配置できないと判定し、命令スケジューリング処理を終了する。

（ステップ S 4 1 3）ステップ S 4 0 7 から繰り返す。

（ステップ S 4 1 4）ステップ S 4 0 3 から繰り返す。

（具体例）

再び、図 1 5 のプログラムを例に、クロックサイクルの目標数を 4 とした場合について説明する。第 3 の実施の形態におけるデータ依存関係解析部 1 4 0 は、図 1 6 に示した従来と同様のデータ依存グラフを生成し、先行制約順位算出部 1 5 1 は、当該データ依存グラフから従来の優先順位である先行制約順位を算出する。

【 0 0 8 1 】

図 1 1 及び図 1 2 は、命令スケジューリング部 4 3 0 による各命令の配置過程を示している。

両図中、命令欄 5 0 1 は各命令を符号で示し、使用資源欄 5 0 2 は当該命令の処理に使用される資源がメモリアクセス部、又は演算部であることを、それぞれ記号 M、A で示し、先行制約順位欄 5 0 3 は、当該命令に算出される先行制約順位を示している。

【 0 0 8 2 】

第 1 決定欄 5 1 0 乃至第 7 決定欄 5 8 0 は、配置の決定順に、それぞれの決定について、配置状態、算出される残サイクル数、算出される資源制約値を示している。配置状態欄について、未配置でかつその回に配置不可、未配置でかつその回に配置可能、及び配置済である状態が、未決定、配置可能、及び決定されたサイクル番号により、それぞれ表されると共に、その回において 1 つの配置可能命令に決定されるサイクル番号が付記される。

【 0 0 8 3 】

配置結果欄 5 9 0 は、最終的に各命令が配置されるサイクル番号を示している。

以下、各回の決定について、詳細に説明する。

（第 1 決定）命令スケジューリング部 4 3 0 は、データ依存関係にある先行命令を持たない命令 A を唯一の配置可能命令と認識し、A のみを要素とするリスト {A} を作成する。命令選択部 1 6 1 は、命令 A を選択し、実行時期決定部 4 6 0 は、命令 A をサイクル 1 に配置し、命令スケジューリング部 4 3 0 は、A を当該リストから除外する。

【 0 0 8 4 】

命令 A が配置された後の配置可能命令は、命令 B、命令 C、及び命令 E の 3 つである。これらの命令は、対象プロセッサにおいて、それぞれ演算部、演算部、及びメモリアクセス部により処理される。この時点で、演算部によって処理される未配置命令は、命令 B、命令 C、及び命令 D の 3 つであり、メモリアクセス部によって処理される未配置命令は、命令 E、命令 F、及び命令 G の 3 つである。

【 0 0 8 5 】

命令 B に対し、資源制約評価部 1 5 2 は、演算部によって処理される未配置命令の数 3 を、演算部が並行して処理可能な最大命令数 2 で除して得られる資源制約値 1. 5 を算出する。

また、決定可否判断部 4 6 2 は、命令 B と真の依存関係にある命令 A が配置されたサイクル 1 よりも後で、配置済の命令数が前記共通最大数に達していないクロックサイクルが、サイクル 2、サイクル 3、及びサイクル 4 の 3 つであることを認識し、残サイクル数 3 を算出する。

【 0 0 8 6 】

同様に、命令 C に対し、資源制約評価部 1 5 2 は、資源制約値 1. 5 を算出し、決定可否判断部 4 6 2 は、残サイクル数 3 を算出する。

命令 E に対し、資源制約評価部 1 5 2 は、メモリアクセス部によって処理される未配置命令の数 3 を、メモリアクセス部が並行して処理可能な最大命令数 1 で除して得られる資源制約値 3 を算出する。

【 0 0 8 7 】

また、決定可否判断部 4 6 2 は、命令 E と真の依存関係にある命令 A が配置されたサイクル 1 よりも後で、配置済の命令数が前記共通最大数に達していないクロックサイクルが、サイクル 2、サイクル 3、及びサイクル 4 の 3 つであることを認識し、残サイクル数 3 を算出する。

何れの命令についても資源制約値が残サイクル数を上回らないので、第 2 決定処理へ進む。

【 0 0 8 8 】

(第 2 決定) 第 2 決定において、命令 B がサイクル 2 に配置される。当該配置後、配置可能である命令 C 及び命令 E について、同様に、資源制約値と残サイクル数とが算出される。何れの命令についても資源制約値が残サイクル数を上回らないので、第 3 決定処理へ進む。

(第 3 決定) 命令スケジューリング部 4 3 0 は、全ての先行命令が配置済となった命令 C 及び命令 E を配置可能命令と認識し、リスト {C、E} を作成する。命令選択部 1 6 1 は、命令 C を選択し、実行時期決定部 4 6 0 は、命令 C をサイ

クル 2 に配置し、命令スケジューリング部 4 3 0 は、C を当該リストから除外する。

【 0 0 8 9 】

命令 C が配置された後の配置可能命令は、命令 D 及び命令 E の 2 つである。これらの命令は、対象プロセッサにおいて、それぞれ演算部及びメモリアクセス部により処理される。この時点で、演算部によって処理される未配置命令は、命令 D のみであり、メモリアクセス部によって処理される未配置命令は、命令 E、命令 F、及び命令 G の 3 つである。

【 0 0 9 0 】

命令 D に対し、資源制約評価部 1 5 2 は、演算部によって処理される未配置命令の数 1 を、演算部が並行して処理可能な最大命令数 2 で除して得られる資源制約値 0.5 を算出する。

また、決定可否判断部 4 6 2 は、命令 D と真の依存関係にある命令 C が配置されたサイクル 2 よりも後において、配置済の命令数が前記共通最大数に達していないクロックサイクルが、サイクル 3、及びサイクル 4 の 2 つであることを認識し、残サイクル数 2 を算出する。

【 0 0 9 1 】

命令 E に対し、資源制約評価部 1 5 2 は、メモリアクセス部によって処理される未配置命令の数 3 を、メモリアクセス部が並行して処理可能な最大命令数 1 で除して得られる資源制約値 3 を算出する。

また、決定可否判断部 4 6 2 は、命令 E と真の依存関係にある命令 A が配置されたサイクル 1 よりも後で、配置済の命令数が前記共通最大数に達していないクロックサイクルが、サイクル 3、及びサイクル 4 の 2 つであることを認識し、残サイクル数 2 を算出する。

【 0 0 9 2 】

ここで、命令 E についても資源制約値が残サイクル数を上回るなので、再決定制御部 4 6 4 は、命令 C について行われた決定を撤回して、他の命令の配置を試みる。

(第 3 決定 - 再試行) 第 3 決定の再試行において、前記リストが {E} となっ

ており、命令 E がサイクル 2 に配置される。

【 0 0 9 3 】

命令 E が配置された後の配置可能命令は、決定を撤回された命令 C、及び命令 F の 2 つである。これらの命令は、対象プロセッサにおいて、それぞれ演算部及びメモリアクセス部により処理される。この時点で、演算部によって処理される未配置命令は、命令 C、及び命令 D の 2 つであり、メモリアクセス部によって処理される未配置命令は、命令 F、及び命令 G の 2 つである。

【 0 0 9 4 】

命令 C に対し、資源制約評価部 1 5 2 は、資源制約値 1 を算出し、決定可否判断部 4 6 2 は、残サイクル数 2 を算出する。

命令 E に対し、資源制約評価部 1 5 2 は、資源制約値 2 を算出し、決定可否判断部 4 6 2 は、残サイクル数 2 を算出する。

何れの命令についても資源制約値が残サイクル数を上回らないので、第 4 決定処理へ進む。

【 0 0 9 5 】

(第 4 決定乃至第 7 決定) 第 4 決定乃至第 7 決定において、決定の再試行は発生しない。図 1 2 に示した決定が各回に行われる。

図 1 3 は、このような配置過程から得られる命令の配置であり、目標の 4 クロックサイクルへの全ての命令の配置に成功している。

各命令が配置されるクロックサイクルは、第 1 及び第 2 の実施の形態における配置 (図 5 参照) と同様であるが、決定の順序が一部異なる。

(第 3 のまとめ)

以上述べたように、第 3 の実施の形態における命令スケジューリング装置は、所定目標数のクロックサイクルに命令の配置を試みる。このために、先行制約順位を用いて各命令を配置し、配置が行われるごとに未配置命令が当該目標数のクロックサイクルに配置できなくなるか否か資源制約を考慮して判断し、否と判断される場合、直前の配置を撤回すると共に他の命令の配置を試みる。

【 0 0 9 6 】

本命令スケジューリング装置は、資源制約を考慮して未配置命令の配置可否判

断を行い、それに応じて配置の再試行を制御するので、同様の判断をデータ依存関係にのみ基づいて行い配置の再試行を制御する方法に比べてより多くの場合、資源制約の厳しい命令を含む全命令の配置に成功する。

(その他の変形例)

なお、本発明を上記の実施の形態に基づいて説明してきたが、本発明は、上記の実施の形態に限定されないのはもちろんである。以下のような場合も本発明に含まれる。

(1) 本発明は、実施の形態で説明したステップを含む方法であるとしてもよい。また、これらの方法を、コンピュータシステムを用いて実現するためのコンピュータプログラムであるとしてもよいし、前記プログラムを表すデジタル信号であるとしてもよい。

【0097】

また、本発明は、前記プログラム又は前記デジタル信号を記録したコンピュータ読取り可能な記録媒体、例えば、フレキシブルディスク、ハードディスク、CD-ROM、MO、DVD、DVD-ROM、DVD-RAM、半導体メモリ等であるとしてもよい。

また、本発明は、電気通信回線、無線又は有線通信回線、若しくはインターネットに代表されるネットワーク等を経由して伝送される前記コンピュータプログラム又は前記デジタル信号であるとしてもよい。

【0098】

また、本発明は、マイクロプロセッサ及びメモリを備えたコンピュータシステムであり、前記メモリは前記プログラムを記憶しており、前記マイクロプロセッサは前記メモリに記憶されている前記プログラムに従って動作することにより、前記方法を実現するとしてもよい。

また、前記プログラム又は前記デジタル信号は、前記記録媒体に記録されて移送され、若しくは、前記ネットワーク等を経由して移送され、独立した他のコンピュータシステムにおいて実施されるとしてもよい。

(2) 本発明を説明するために例示したプログラム(図15参照)は、ソースプログラムをコンパイルして得られた並列最適化前のプログラムの総体であっても

よく、また、そのようなプログラムの一つの基本ブロックに対応する部分であってもよい。これらの何れの場合も、本発明に含まれる。

(3) 第3の実施の形態において、ステップS411で直前の命令の配置が撤回された場合、ステップS405へ戻ってリスト中の他の命令の配置を試行し、リスト中の各命令の配置に失敗した場合、ステップS412で目標クロックサイクルへの命令の配置はできないと判定するとした。

【0099】

しかしながら、別法として、ステップS404で過去に生成された配置可能命令のリストを記録すると共に、現在のリスト中の各命令の配置の再試行に失敗した場合、直ちに目標クロックサイクルへ配置不能と判定せず、過去のリストから決定された命令の配置を撤回し、その過去のリスト中の他の命令の配置を試行してもよい。

【0100】

このような処理は、いわゆるバックトラックアルゴリズムとして従来一般化されており、当該アルゴリズムの一具体化として容易に実施可能である。過去のリストを記録すると共に、あるリスト中の各命令の配置に失敗した場合に、過去のリストをバックトラックする場合も、本発明に含まれる。

【0101】

【発明の効果】

本発明の命令スケジューリング方法は、対象プログラム中に記述された複数命令のうち1つ以上の命令を対象命令とし、各対象命令に、自命令を処理するハードウェア資源固有の制約条件と、当該ハードウェア資源によって処理される他命令との間に存するデータ依存関係とに基づいて優先順位を算出する優先順位算出ステップと、最も高い優先順位を算出された対象命令の実行時期を決定する実行時期決定ステップとを含む。

【0102】

この方法によれば、ハードウェア資源固有の制約条件が考慮された前記優先順位を命令の選択と配置に用いることにより、資源制約の厳しい命令を優先的に早いクロックサイクルに配置できる。これにより、そのような命令を含む全命令を

、従来よりも少ないクロックサイクルに配置できる。

また、前記命令スケジューリング方法において、前記優先順位算出ステップは、前記各対象命令に対し、前記データ依存関係にのみ基づいて先行制約順位を算出する先行制約順位算出サブステップと、前記各対象命令が、自命令と同一の先行制約順位を算出され、かつ前記データ依存関係にある後続命令を持つ場合、自命令とその後続命令とを前記ハードウェア資源が並行して処理可能か否かを判断する資源制約評価サブステップとを含み、否と判断された各対象命令に対し、前記算出された先行制約順位を繰り上げて得られる優先順位を算出し、その他の各対象命令に対し、前記算出された先行制約順位と等しい優先順位を算出してもよい。

【 0 1 0 3 】

この方法によれば、データ依存関係にあって、かつ同一の先行制約順位を算出される2つの命令が、対象プロセッサにおけるハードウェア資源によって並行処理されない場合、その先行命令の優先順位を従来の先行制約順位よりも高く算出するので、従来の技術では発見されなかった資源制約のために生じるクリティカルパスを発見することができる。

【 0 1 0 4 】

そして、そのようなクリティカルパス上の先頭命令をできるだけ早いクロックサイクルに配置するので、資源制約のために並行処理されない命令を含む全命令を、従来よりも少ないクロックサイクルに配置できる。

また、前記命令スケジューリング方法において、前記優先順位算出ステップは、前記各対象命令に対し、（１）前記データ依存関係にある後続命令を持たない場合１、（２）逆依存関係又は出力依存関係にある１つ以上の後続命令を持つ場合、当該後続命令に算出された最大値、及び（３）真の依存関係にある１つ以上の後続命令を持つ場合、当該後続命令に算出された最大値よりも１大きい値を、それぞれ先行制約順位として算出する先行制約順位算出サブステップと、前記各対象命令に対し、自命令を処理するハードウェア資源によって処理され実行時期未決定なる命令の総数を、当該ハードウェア資源が並行して処理可能な最大命令数で除して得られる資源制約値を算出する資源制約評価サブステップとを含み、

当該資源制約値が当該先行制約順位よりも大きい場合、前記対象命令に当該資源制約値を優先順位として算出し、その他の場合、前記対象命令に当該先行制約順位を優先順位として付与し、前記実行時期決定ステップは、最も大きい優先順位を算出された対象命令の実行時期を決定してもよい。

【 0 1 0 5 】

この方法によれば、前記各対象命令に、資源制約値と先行制約順位の何れか大きい方を優先順位として算出し、最も高い優先順位を算出された命令を選択して配置することにより、資源制約の厳しい命令を従来よりも早いクロックサイクルに配置することができるので、そのような命令を含む全命令を従来よりも少ないクロックサイクルに配置できる。

【 0 1 0 6 】

特に、並列処理可能な命令数の小さな資源によって処理される多数の未配置命令があって、それらがデータ依存関係にない場合に、そのような命令に大きな資源制約値が算出されるので、早いクロックサイクルに適切に配置されるという特有の効果が生じる。

本発明の命令スケジューリング方法は、対象プログラム中に記述された複数命令の実行時期をそれぞれ決定する命令スケジューリング方法であって、第 1 命令の実行時期を決定した場合、第 2 命令の実行時期を所定期間に含まれるように決定できなくなるか否かを、当該第 2 命令を処理するハードウェア資源固有の制約条件に基づいて判断する決定可否判断ステップと、当該決定ができなくなると判断された場合、当該第 1 命令の実行時期の決定を撤回すると共に、当該第 1 命令と異なる命令の実行時期を決定する再決定ステップとを含む。

【 0 1 0 7 】

また、前記命令スケジューリング方法において、前記所定期間は、クロックサイクル数により示され、前記決定可否判断ステップは、前記第 2 命令を処理するハードウェア資源によって処理され実行時期未決定なる命令の総数を、当該ハードウェア資源が並行して処理可能な最大命令数で除して得られる資源制約値を算出する資源制約評価サブステップを含み、当該資源制約値が前記クロックサイクル数よりも大きければ、前記決定ができなくなると判断してもよい。

【 0 1 0 8 】

これらの方法によれば、命令の実行時期を所定期間に含まれるように決定できなくなるか否か資源制約を考慮して判断し、否と判断される場合、直前の配置を撤回すると共に他の命令の配置を行うので、同様の判断をデータ依存関係にのみ基づいて行い配置の撤回を制御する方法に比べてより多くの場合、資源制約の厳しい命令を含む全命令の配置に成功する。

【 0 1 0 9 】

本発明のプログラム変換方法は、入力プログラムを、複数命令によって記述された中間プログラムに変換し、当該中間プログラムに対して前述の何れかの命令スケジューリング方法を適用することにより、当該複数命令の実行時期を決定することを特徴とする。

この方法によれば、前述の命令スケジューリング方法を当該中間プログラムに適用するので、従来よりも高度に並列最適化された目的プログラムが生成可能となる。

【 0 1 1 0 】

本発明の命令スケジューリング装置は、対象プログラム中に記述された複数命令のうち1つ以上の命令を対象命令とし、各対象命令に、自命令を処理するハードウェア資源固有の制約条件と、当該ハードウェア資源によって処理される他命令との間に存するデータ依存関係とに基づいて優先順位を算出する優先順位算出手段と、最も高い優先順位を算出された対象命令の実行時期を決定する実行時期決定手段とを備える。

【 0 1 1 1 】

本発明の命令スケジューリング装置は、対象プログラム中に記述された複数命令の実行時期をそれぞれ決定する命令スケジューリング装置であって、第1命令の実行時期を決定した場合、第2命令の実行時期を所定期間に含まれるように決定できなくなるか否かを、当該第2命令を処理するハードウェア資源固有の制約条件に基づいて判断する決定可否判断手段と、当該決定ができなくなると判断された場合、当該第1命令の実行時期の決定を撤回すると共に、当該第1命令と異なる命令の実行時期を決定する再決定手段とを備える。

【 0 1 1 2 】

これらの構成によれば、前述と同様効果を有する命令スケジューリング装置を実現できる。

本発明のプログラムは、命令スケジューリングをコンピュータを用いて行うための、コンピュータ実行可能なプログラムであって、対象プログラム中に記述された複数命令のうち1つ以上の命令を対象命令とし、各対象命令に、自命令を処理するハードウェア資源固有の制約条件と、当該ハードウェア資源によって処理される他命令との間に存するデータ依存関係とに基づいて優先順位を算出する優先順位算出ステップと、最も高い優先順位を算出された対象命令の実行時期を決定する実行時期決定ステップとを前記コンピュータに実行させる。

【 0 1 1 3 】

本発明のプログラムは、対象プログラム中に記述された複数命令の実行時期を、コンピュータを用いてそれぞれ決定するための、コンピュータ実行可能なプログラムであって、第1命令の実行時期を決定した場合、第2命令の実行時期を所定期間に含まれるように決定できなくなるか否かを、当該第2命令を処理するハードウェア資源固有の制約条件に基づいて判断する決定可否判断ステップと、当該決定ができなくなると判断された場合、当該第1命令の実行時期の決定を撤回すると共に、当該第1命令と異なる命令の実行時期を決定する再決定ステップとを前記コンピュータに実行させる。

【 0 1 1 4 】

これらのプログラムによれば、前述と同様の効果を有する命令スケジューリング処理をコンピュータに実行させることができる。

本発明のプログラム記録媒体は、コンピュータ読み取り可能な記録媒体であって、前述のプログラムを記録していることを特徴とする。

このプログラム記録媒体によれば、前述した効果を有するプログラムを移送し所期のコンピュータにおいて実行させることができる。

【図面の簡単な説明】

【図 1】

コンパイラ装置 1 0 0 の全体構成を示す機能ブロック図である。

【図 2】

コンパイラ装置 1 0 0 が対象とするプロセッサの一構成例である。

【図 3】

命令スケジューリング処理を示すフローチャートである。

【図 4】

データ依存関係解析部 1 4 0 によって生成されるデータ依存グラフの一例である。

【図 5】

命令の配置の一例である。

【図 6】

命令スケジューリング処理を示すフローチャートである。

【図 7】

各命令の配置過程の一例である。

【図 8】

各命令の配置過程の一例である。

【図 9】

コンパイラ装置 4 0 0 の全体構成を示す機能ブロック図である。

【図 1 0】

命令スケジューリング処理を示すフローチャートである。

【図 1 1】

各命令の配置過程の一例である。

【図 1 2】

各命令の配置過程の一例である。

【図 1 3】

命令の配置の一例である。

【図 1 4】

従来装置が行う命令スケジューリング処理の一例を示すフローチャートである。

【図 1 5】

従来装置に入力される入力プログラムの一例である。

【図 1 6】

入力プログラムに対して従来装置が生成するデータ依存グラフである。

【図 1 7】

入力プログラムの各命令が配置されるクロックサイクルの一例である。

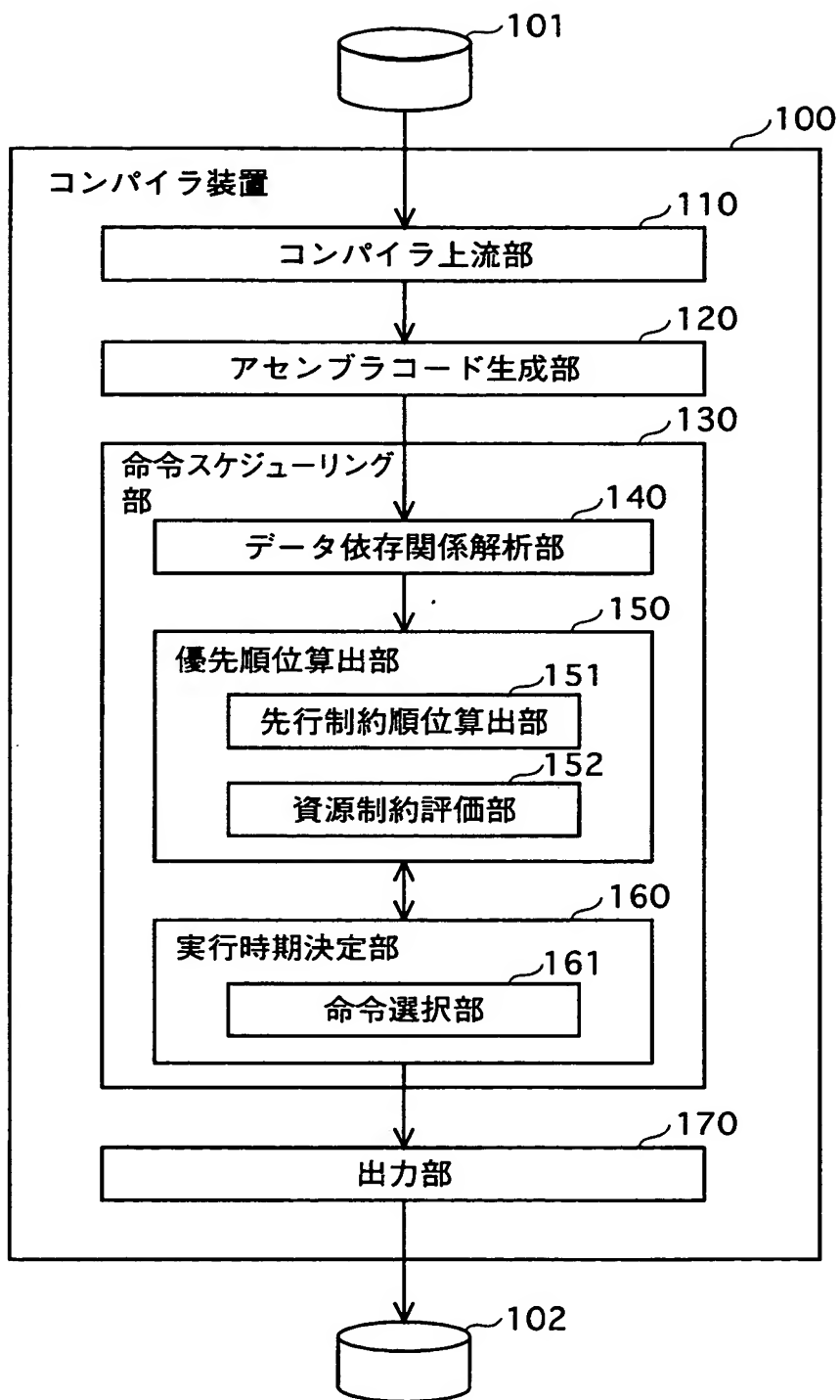
【符号の説明】

1 0 0	コンパイラ装置
1 0 1	ソースファイル
1 0 2	目的ファイル
1 1 0	コンパイラ上流部
1 2 0	アセンブラコード生成部
1 3 0	命令スケジューリング部
1 4 0	データ依存関係解析部
1 5 0	優先順位算出部
1 5 1	先行制約順位算出部
1 5 2	資源制約評価部
1 6 0	実行時期決定部
1 6 1	命令選択部
1 7 0	出力部
3 0 1	命令欄
3 0 2	使用資源欄
3 0 3	先行制約順位欄
3 1 0 ~ 3 7 0	第 1 決定欄 ~ 第 7 決定欄
3 8 0	配置結果欄
4 0 0	コンパイラ装置
4 3 0	命令スケジューリング部
4 6 0	実行時期決定部
4 6 2	決定可否判断部
4 6 4	再決定制御部
5 0 1	命令欄

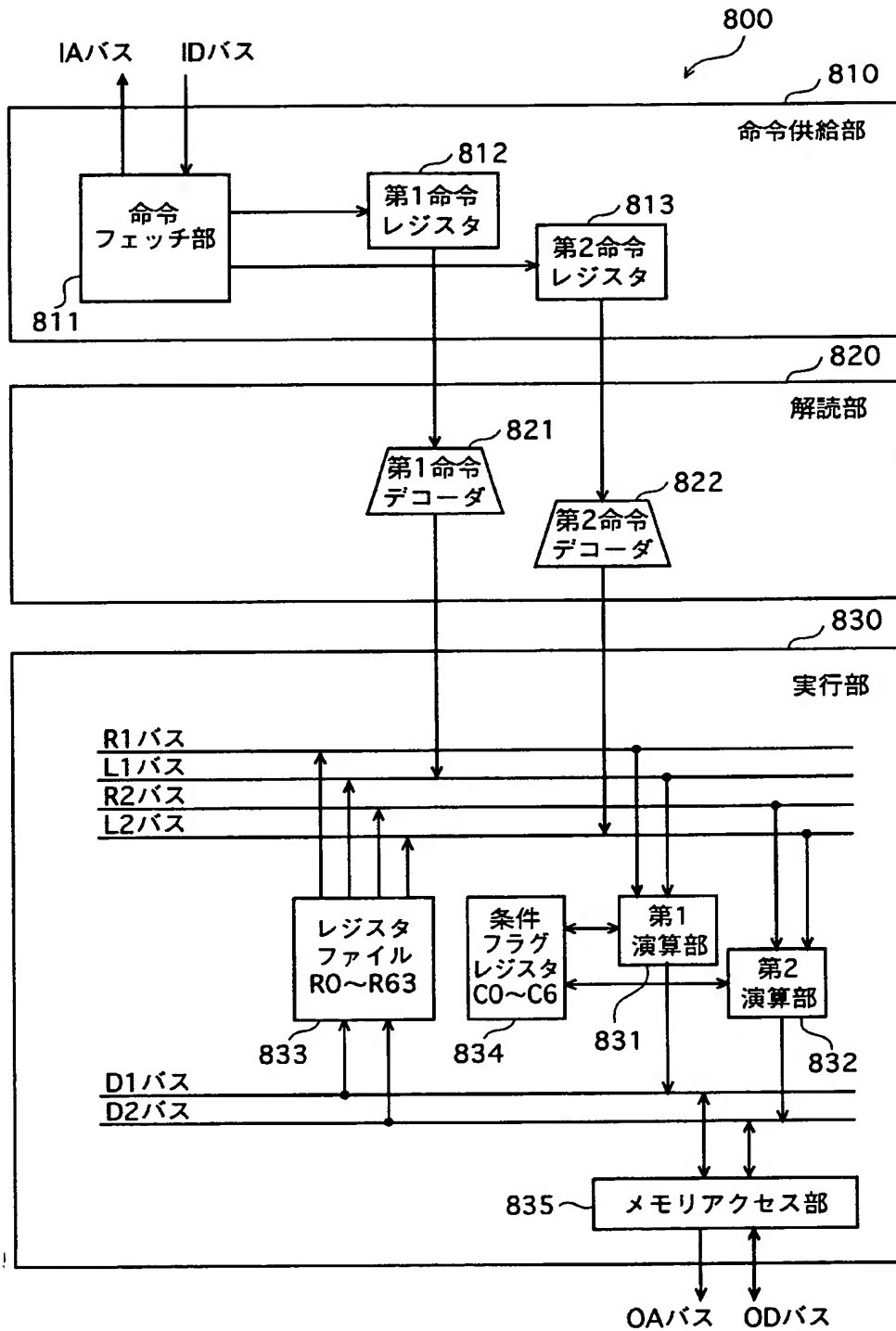
5 0 2	使用資源欄
5 0 3	先行制約順位欄
5 1 0 ~ 5 8 0	第 1 決定欄 ~ 第 7 決定欄
5 9 0	配置結果欄
8 0 0	プロセッサ
8 1 0	命令供給部
8 1 1	命令フェッチ部
8 1 2	第 1 命令レジスタ
8 1 3	第 2 命令レジスタ
8 2 0	解読部
8 2 1	第 1 命令デコーダ
8 2 2	第 2 命令デコーダ
8 3 0	実行部
8 3 1	第 1 演算部
8 3 2	第 2 演算部
8 3 3	レジスタファイル
8 3 4	条件フラグレジスタ
8 3 5	メモリアクセス部
9 0 1	クロックサイクル欄
9 0 2	命令 1 欄
9 0 3	命令 2 欄

【書類名】 図面

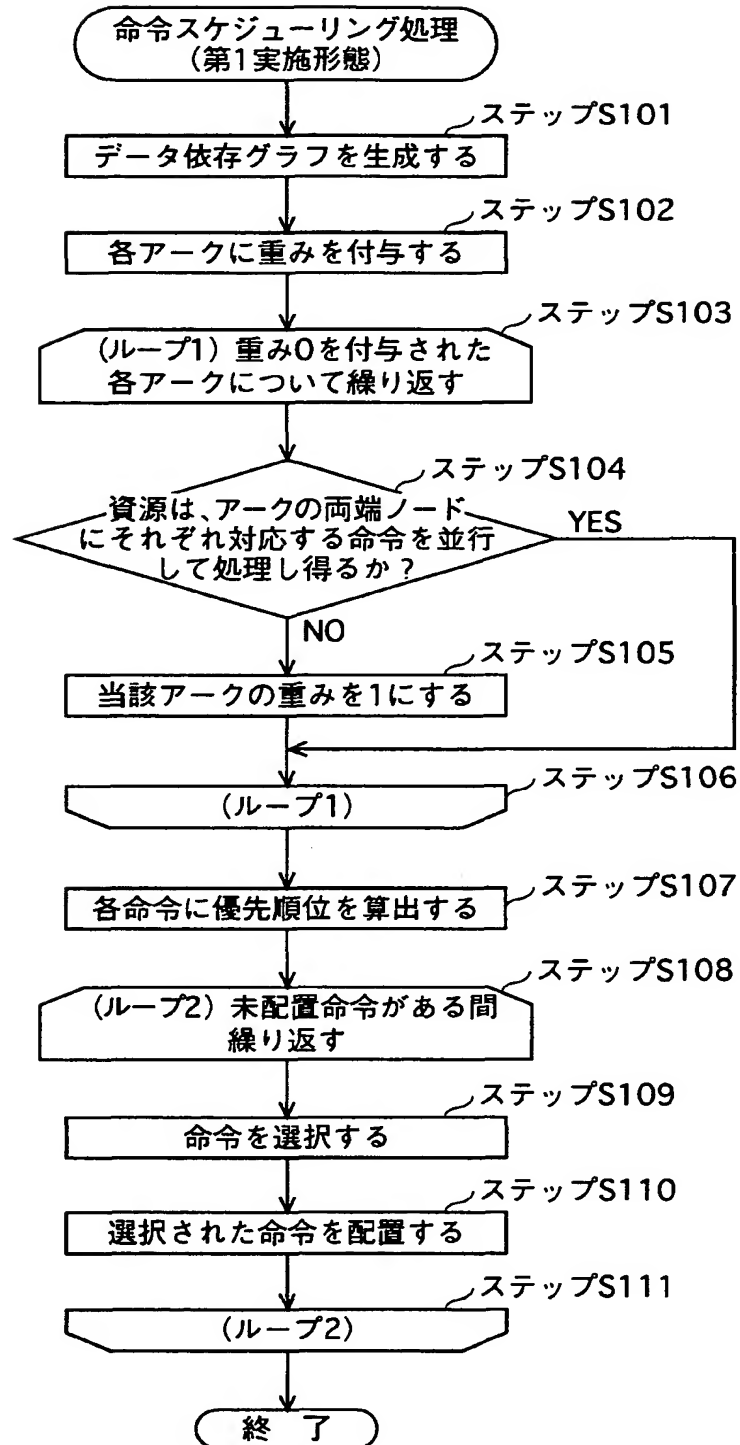
【図 1】



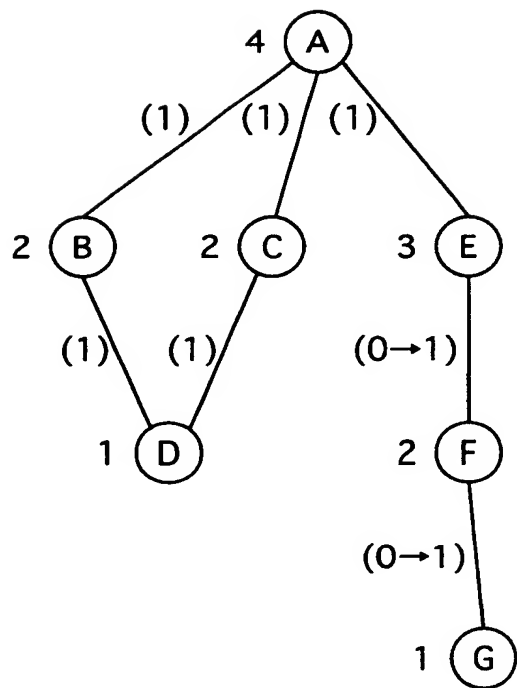
【図2】



【図 3】



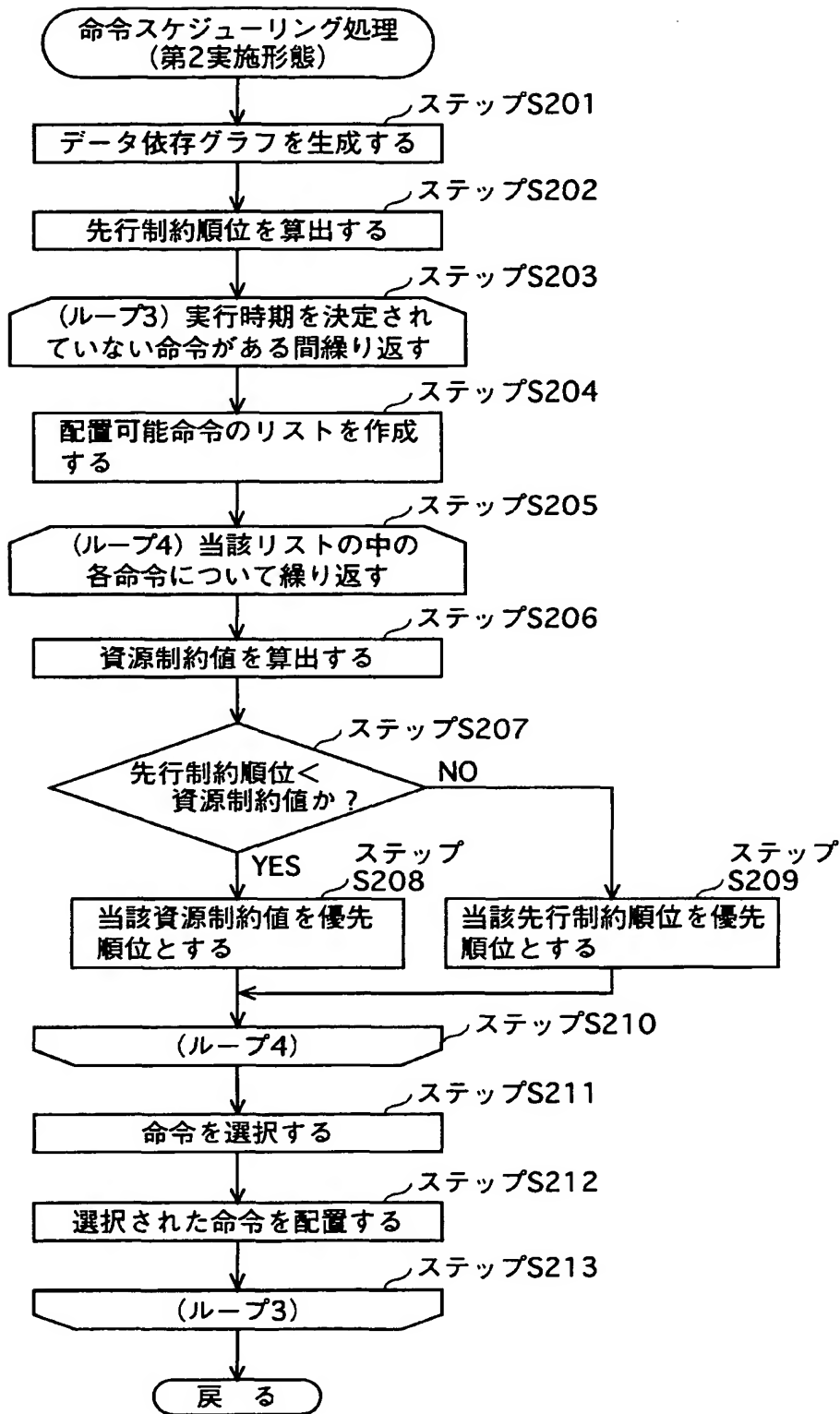
【図 4】



【図 5】

クロックサイクル	命令1	命令2
1	ld r0, (r10) ... ① (第1決定)	なし
2	st (r11), r0 ... ② (第2決定)	add r3, r0, r1 ... ③ (第3決定)
3	add r4, r0, r2 ... ④ (第4決定)	ld r0, (r12) ... ⑤ (第5決定)
4	or r5, r4, r3 ... ⑥ (第6決定)	ld r0, (r13) ... ⑦ (第7決定)

【図 6】



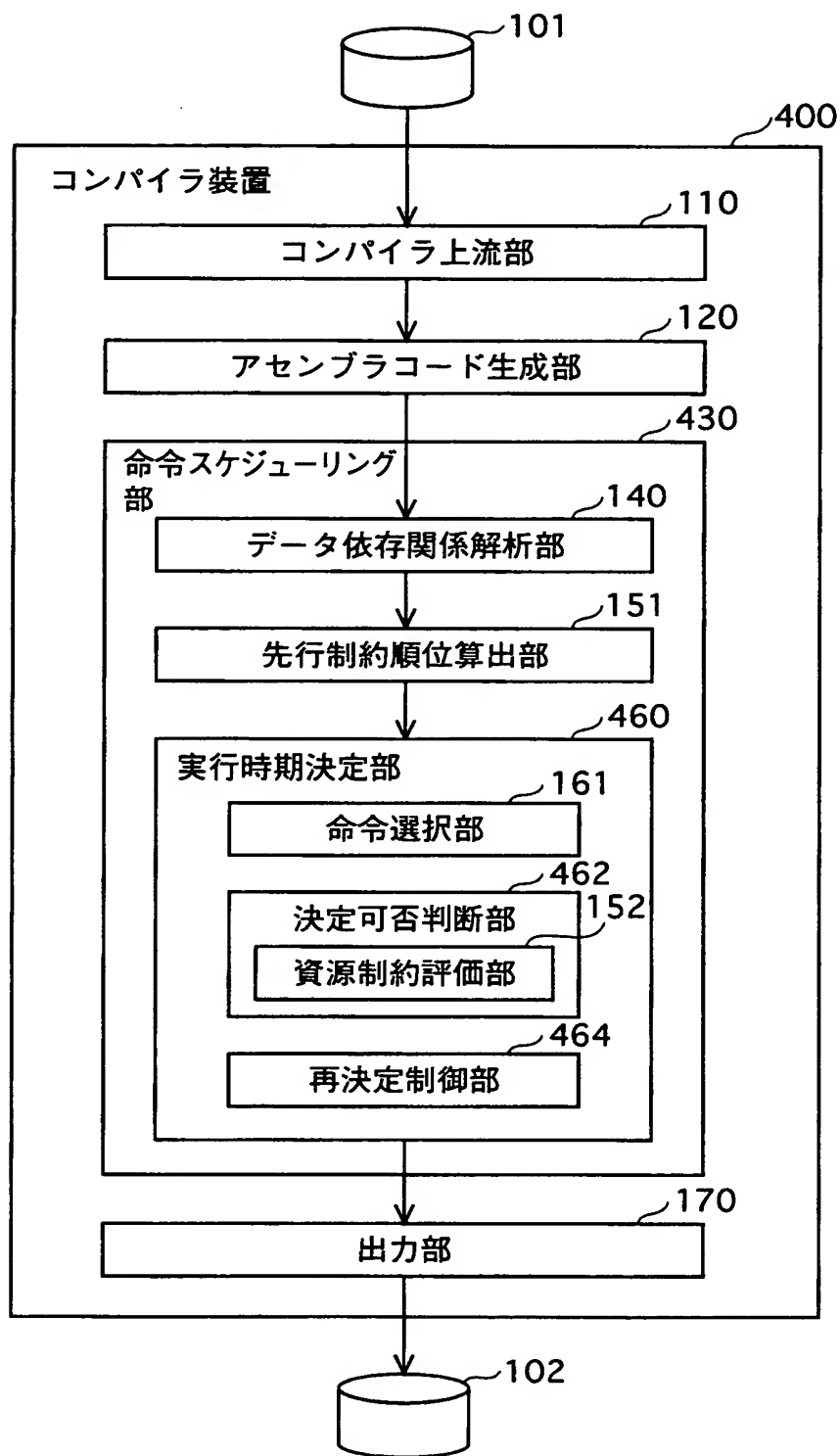
【図 7】

命令	使用 資源	先行 制約 順位	310 第1決定			320 第2決定			330 第3決定			340 第4決定		
			配置状態	資源 制約値	優先順位	配置状態	資源 制約値	優先順位	配置状態	資源 制約値	優先順位	配置状態	資源 制約値	優先順位
A	M	3	配置可能	4	4	1	-	-	1	-	-	1	-	-
B	A	2	未決定	-	-	配置可能	1.5	2	配置可能	1.5	2	2	-	-
C	A	2	未決定	-	-	配置可能	1.5	2	配置可能	1.5	2	配置可能	1	2
D	A	1	未決定	-	-	未決定	-	-	未決定	-	-	未決定	-	-
E	M	1	未決定	-	-	配置可能	3	3	2	-	-	2	-	-
F	M	1	未決定	-	-	未決定	-	-	配置可能	2	2	配置可能	2	2
G	M	1	未決定	-	-	未決定	-	-	未決定	-	-	未決定	-	-

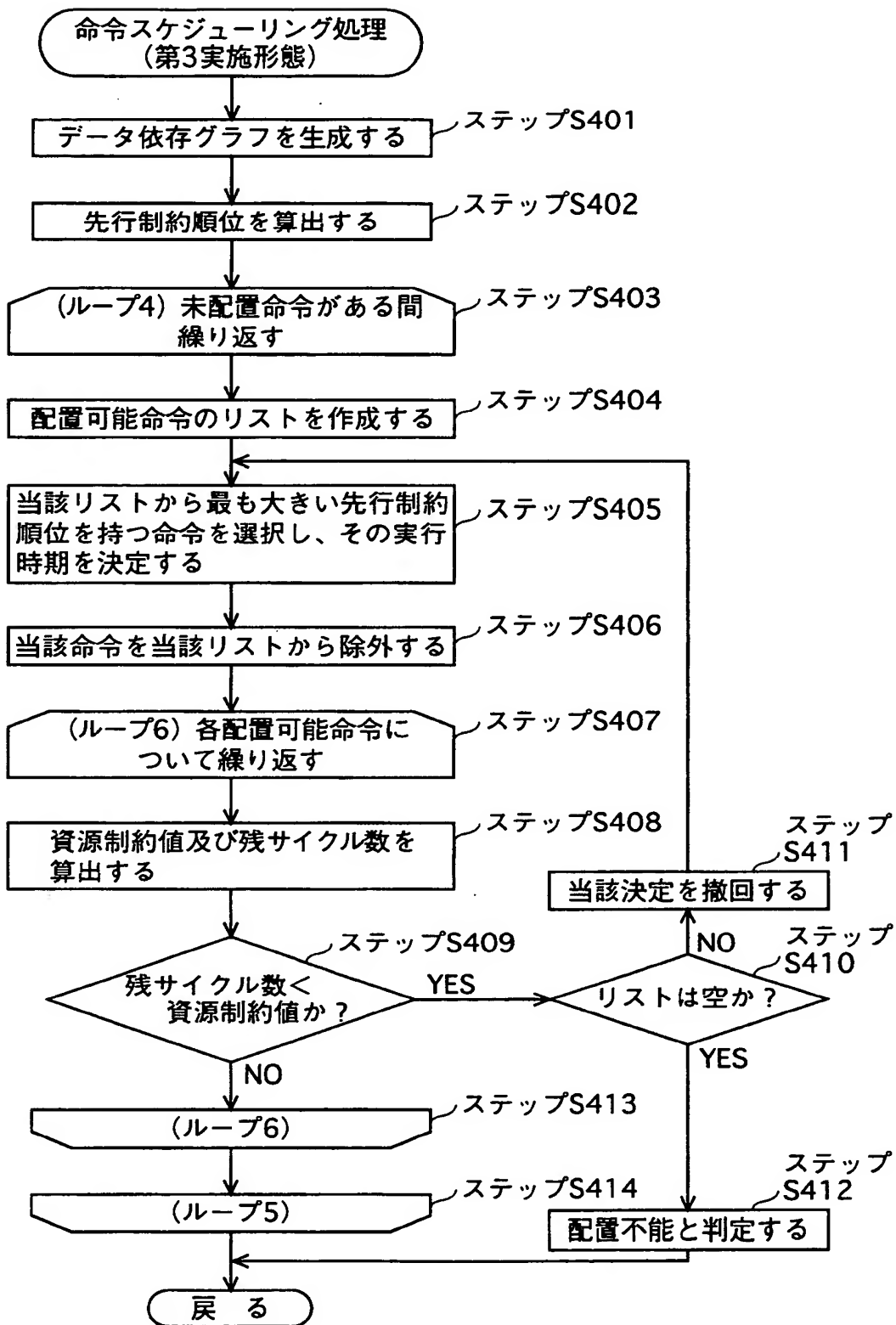
【図 8】

命令	使用資源	先行制約順位	350 第5決定				360 第6決定				370 第7決定				配置結果
			配置狀態	資源制約值	優先順位	配置狀態	資源制約值	優先順位	配置狀態	資源制約值	優先順位				
A	M	3	1	-	-	1	-	-	1	-	-	1	-	1	
B	A	2	2	-	-	2	-	-	2	-	-	2	-	2	
C	A	2	3	-	-	3	-	-	3	-	-	3	-	3	
D	A	1	配置可能	0.5	1	配置可能	0.5	1	4	-	-	4	-	4	
E	M	1	2	-	-	2	-	-	2	-	-	2	-	2	
F	M	1	配置可能	2	2	3	-	-	3	-	-	3	-	3	
G	M	1	未決定	-	-	配置可能	1	1	配置可能	1	1	配置可能	1	4	

【図 9】



【図10】



【図 1 1】

命令	使用 資源	先行 制約 順位	510 第1決定			520 第2決定			530 第3決定			540 第3決定(再決定)		
			配置状態	残サイ クル数	資源 制約値	配置状態	残サイ クル数	資源 制約値	配置状態	残サイ クル数	資源 制約値	配置状態	残サイ クル数	資源 制約値
A	M	3	配置可能 1に決定	-	-	1	-	-	1	-	-	1	-	-
B	A	2	未決定	3	1.5	配置可能 2に決定	-	-	2	-	-	2	-	-
C	A	2	未決定	3	1.5	配置可能	3	1	配置可能 2に決定	-	-	配置可能 決定撤回	2	1
D	A	1	未決定	-	-	未決定	-	-	未決定	2	0.5	未決定	-	-
E	M	1	未決定	3	3	配置可能	3	3	配置可能	2	3	配置可能 2に決定	-	-
F	M	1	未決定	-	-	未決定	-	-	未決定	-	-	未決定	2	2
G	M	1	未決定	-	-	未決定	-	-	未決定	-	-	未決定	-	-

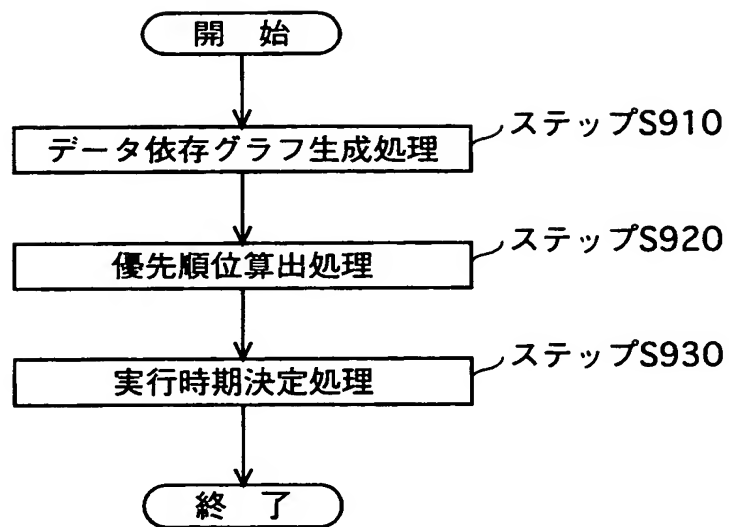
【図12】

命令	使用資源	先行制約順位	550 第4決定			560 第5決定			570 第6決定			580 第7決定			590 配置結果
			配置状態	残サイクル数	資源制約値	配置状態	残サイクル数	資源制約値	配置状態	残サイクル数	資源制約値	配置状態	残サイクル数	資源制約値	
A	M	3	1	-	-	1	-	-	1	-	-	1	-	-	1
B	A	2	2	-	-	2	-	-	2	-	-	2	-	-	2
C	A	2	配置可能3に決定	-	-	3	-	-	3	-	-	3	-	-	3
D	A	1	未決定	1	0.5	配置可能4に決定	-	-	4	-	-	4	-	-	4
E	M	1	2	-	-	2	-	-	2	-	-	2	-	-	2
F	M	1	配置可能	2	2	配置可能	2	2	配置可能3に決定	-	-	3	-	-	3
G	M	1	未決定	-	-	未決定	-	-	未決定	1	1	配置可能4に決定	-	-	4

【図 1 3】

クロックサイクル	命令1	命令2
1	ld r0, (r10) ... ㉠ (第1決定)	なし
2	add r3, r0, r1 ... ㉢ (第2決定)	st (r11), r0 ... ㉤ (第3決定)
3	add r4, r0, r2 ... ㉣ (第4決定)	ld r0, (r12) ... ㉦ (第6決定)
4	or r5, r4, r3 ... ㉤ (第5決定)	ld r0, (r13) ... ㉧ (第7決定)

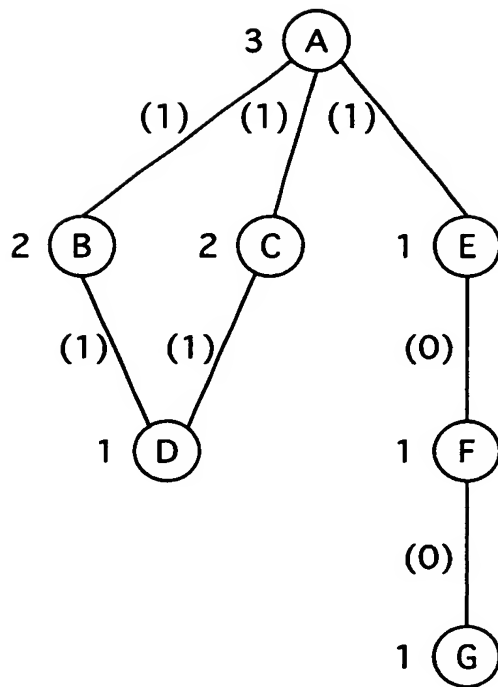
【図 1 4】



【図 1 5】

ld	r0, (r10)	. . .	Ⓐ
add	r3, r0, r1	. . .	Ⓑ
add	r4, r0, r2	. . .	Ⓒ
or	r5, r4, r3	. . .	Ⓓ
st	(r11), r0	. . .	Ⓔ
ld	r0, (r12)	. . .	Ⓕ
ld	r0, (r13)	. . .	Ⓖ

【図 1 6】



【図 1 7】

クロックサイクル	命令1	命令2
1	ld r0, (r10) ... ㉠ (第1決定)	なし
2	add r3, r0, r1 ... ㉢ (第2決定)	add r4, r0 r2 ... ㉣ (第3決定)
3	or r5, r4, r3 ... ㉤ (第4決定)	st (r11), r0 ... ㉥ (第5決定)
4	ld r0, (r12) ... ㉦ (第6決定)	なし
5	ld r0, (r13) ... ㉧ (第7決定)	なし

【書類名】 要約書

【要約】

【課題】 本発明は、従来よりも少ないクロックサイクルに命令を配置する命令スケジューリング方法、及び装置を提供する。

【解決手段】 データ依存関係解析部 1 4 0 は、アセンブラコード生成部 1 2 0 から取得した各命令のデータ依存関係を表すデータ依存グラフを生成する。先行制約順位算出部 1 5 1 は、当該グラフの各アークに所定の重み値を付し累計することにより先行制約順位を算出する。資源制約評価部 1 5 2 は、データ依存関係にあってかつ同一の先行制約順位を算出された 2 命令が、資源制約のために並行処理されない場合、その先行命令の先行制約順位を繰上げる。優先順位算出部 1 5 0 は、繰上げ後の先行制約順位を本発明の優先順位とし、命令選択部 1 6 1 は最高優先順位の命令を選択し、実行時期決定部 1 6 0 は選択された命令を配置する処理を、全ての命令が配置済となるまで繰り返す。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 5 8 2 1]

1. 変更年月日 1 9 9 0 年 8 月 2 8 日

[変更理由] 新規登録

住 所 大阪府門真市大字門真 1 0 0 6 番地

氏 名 松下電器産業株式会社